1. **Алгоритм** — это чётко определённая последовательность действий или инструкций, предназначенная для решения определённой задачи или класса задач.

Алгоритм нахождения среднего значения:

1. Определить набор чисел

2. Вычислить сумму всех чисел в наборе

3. Подсчитать количество чисел в наборе

4. Разделить сумму на количество чисел

2. **В языке Python есть несколько основных типов данных**, которые относятся к разным группам: числовым, строковым, последовательностям и словарям. Python — язык с динамической типизацией, что означает, что тип переменной определяется во время выполнения программы, а не на этапе её написания или компиляции.

### Числовые

* **int** — целые числа (положительные или отрицательные, включая нуль). Используются для представления дискретных значений, таких как количество элементов в списке или индекс массива.
* **float** — числа с плавающей точкой (дробная часть, разделённая десятичной точкой). Применяются для представления значений, которые требуют высокой точности, например, измерений в физике или финансовых расчётов.
* **complex** — комплексные числа (имеют реальную и мнимую части). В Python записываются в виде a + bj, где a — действительная часть, а b — мнимая, обозначенная с помощью буквы j (вместо привычной математической i).

### Строковые

* **str** — строковый тип данных для хранения текстовой информации. Строка может содержать буквы, цифры, пробелы, символы и даже эмодзи. [dzen.ru](https://dzen.ru/a/aDcchszqVF6e_Tb3)[base-python.readthedocs.io](https://base-python.readthedocs.io/en/latest/chapters/02_type/03_str/type_str.html)
* **Особенности**:
  + Неизменяемый тип — после создания строку нельзя изменить напрямую, любые операции над ней создают новую строку. [dzen.ru](https://dzen.ru/a/aDcchszqVF6e_Tb3)
  + Поддерживает индексацию и срезы, что позволяет извлекать отдельные символы или подстроки.

### Последовательности

* **Список (list)** — упорядоченная изменяемая последовательность элементов. Определяется с использованием квадратных скобок, элементы разделяются запятыми.
* **Кортеж (tuple)** — упорядоченная неизменяемая коллекция элементов. Определяется с использованием круглых скобок (), элементы кортежа разделяются запятыми.
* **Множество (set)** — неупорядоченная коллекция уникальных элементов. Определяется с использованием фигурных скобок {} (только не пустое), элементы множества разделяются запятыми.

### Словари

* **Словарь (dict)** — изменяемая коллекция, в которой данные хранятся в виде пар «ключ-значение». В отличие от списков, доступ к элементам осуществляется не по индексу, а по ключу.
* **Особенности**:
  + В качестве ключей могут использоваться только хешируемые объекты: строки (str), числа (int, float), кортежи (tuple), если они состоят только из неизменяемых объектов.
  + Значения, в отличие от ключей, могут быть любыми: числа, строки, списки, множества и даже другие словари, функции, объекты классов и любые другие структуры.

3.

**База данных** — это **набор упорядоченных и структурированных данных**, которые хранятся на определённом компьютере. Проще всего представить её как большую Excel-таблицу, где у каждого элемента (строки) есть определённые свойства (столбцы). [skillbox.ru](https://skillbox.ru/media/code/sistema-upravleniya-bazami-dannykh-chto-eto-takoe-i-zachem-ona-nuzhna/)

**Система управления базами данных (СУБД)** — это **набор инструментов**, которые позволяют удобно управлять базами данных: удалять, добавлять, фильтровать и находить элементы, менять их структуру и создавать резервные копии. [skillbox.ru](https://skillbox.ru/media/code/sistema-upravleniya-bazami-dannykh-chto-eto-takoe-i-zachem-ona-nuzhna/)

Если провести аналогию, **база данных — это библиотека с книгами**, а **СУБД — библиотекарь**, который помогает найти нужную книгу и поддерживает порядок на полках. [yandex.cloud](https://yandex.cloud/ru/blog/posts/2025/02/databases)

**Основные функции СУБД**: создание баз данных, изменение, удаление и объединение их по определённым признакам; хранение данных в структурированном виде и нужном формате; защита данных от взлома и нежелательных изменений; выгрузка и сортировка данных по заданным фильтрам; поддержка целостности баз данных, резервное копирование и восстановление после сбоёв.

**Назначение первичного ключа в таблице реляционной базы данных** — **идентификация записей**. Он обеспечивает уникальность каждой строки, не позволяя иметь несколько записей с одинаковыми значениями.

**Реляционная и нереляционная базы данных (NoSQL) отличаются структурой данных, языком запросов, масштабируемостью и производительностью**.

**Реляционные базы данных** хранят структурированные данные в таблицах с предопределёнными связями. Каждая таблица состоит из строк и столбцов, где строки представляют записи, а столбцы — параметры, описывающие каждую запись. [appmaster.io](https://appmaster.io/ru/blog/reliatsionnye-i-nereliatsionnye-bazy-dannykh)[Skyeng.ru](https://Skyeng.ru/it-industry/it/osnovnyye-elementy-struktury-bazy-dannykh/)[skillbox.ru](https://skillbox.ru/media/code/sql_i_nosql_in_i_yan_v_mire_baz_dannykh/)

* **Нереляционные базы данных** не ограничиваются табличной структурой и позволяют хранить данные в разнообразных форматах, таких как документы, графы или «ключ-значение». Структура данных в таких базах динамическая и может меняться в любой момент времени.

4.

**Оперативная (ОЗУ) и постоянная (ПЗУ) память компьютера отличаются назначением и свойствами**. Эти типы памяти выполняют различные функции, и их характеристики отличаются. [guru99.com](https://www.guru99.com/ru/difference-between-rom-ram.html)[struchkov.dev](https://struchkov.dev/blog/ru/storing-data-in-a-computer/)[digitalocean.ru](https://digitalocean.ru/n/chto-takoe-pzu)

### Оперативная память (ОЗУ)

**Назначение**: временное хранилище данных, которое обеспечивает быстрый доступ к информации для процессора. В ОЗУ хранятся данные, которые процессору могут понадобиться в ближайшее время: программный код, данные приложений, информация о состоянии системы и другое. [hi-tech.mail.ru](https://hi-tech.mail.ru/review/125990-operativnaya-pamyat/)[digitalocean.ru](https://digitalocean.ru/n/chto-takoe-operativnaya-pamyat-ozu)[struchkov.dev](https://struchkov.dev/blog/ru/storing-data-in-a-computer/)

**Особенности**:

* **Энергозависима**: при выключении устройства все данные из ОЗУ стираются. [hi-tech.mail.ru](https://hi-tech.mail.ru/review/125990-operativnaya-pamyat/)[skillbox.ru](https://skillbox.ru/media/code/operativnaya-pamyat-ozu-chto-eto-dlya-chego-ona-nuzhna-i-kakaya-byvaet/)
* **Доступ к данным**: центральный процессор может считывать, записывать или изменять данные в оперативной памяти. [tutorialspoint.com](https://tr-page.yandex.ru/translate?lang=en-ru&url=https%3A%2F%2Fwww.tutorialspoint.com%2Fdifference-between-ram-and-rom)
* **Объём**: определяет, сколько данных может хранить ОЗУ одновременно. [hi-tech.mail.ru](https://hi-tech.mail.ru/review/125990-operativnaya-pamyat/)
* **Тактовая частота**: измеряется в МГц, чем выше частота, тем быстрее передаются данные. [hi-tech.mail.ru](https://hi-tech.mail.ru/review/125990-operativnaya-pamyat/)

### Постоянная память (ПЗУ)

**Назначение**: постоянное хранилище данных, которые должны надёжно сохраняться даже при отключении питания устройства. В ПЗУ обычно хранится микропрограмма или загрузчик, необходимый для начальной загрузки системы и выполнения основных операций. [digitalocean.ru](https://digitalocean.ru/n/chto-takoe-pzu)

**Особенности**:

* **Энергонезависима**: сохраняет данные даже при выключении питания.
* **Доступ к данным**: в основном только для чтения, традиционно данные записываются в процессе производства и не могут быть изменены впоследствии (хотя некоторые современные типы, например EEPROM, могут быть перезаписаны).
* **Объём**: меньше по объёму, так как содержит только основные программы и данные, необходимые для работы системы.

5. **Операционная система** — это программное обеспечение, которое управляет ресурсами компьютера и обеспечивает взаимодействие между пользователями и аппаратным обеспечением. [Skyeng.ru](https://Skyeng.ru/magazine/chto-takoe-operacionnaya-sistema/)

**Три примера операционных систем**:

1. **Windows**. Платформа, разработанная корпорацией Microsoft, которая широко применяется на персональных компьютерах различных производителей.
2. **macOS**. Операционная система, созданная компанией Apple, которая является неотъемлемой частью всех устройств этого производителя.
3. **Linux**. Семейство свободных операционных систем, основанных на ядре Linux, и популярна среди пользователей, предпочитающих открытое программное обеспечение.

[Skyeng.ru](https://Skyeng.ru/magazine/chto-takoe-operacionnaya-sistema/)[Trends.RBC.ru](https://Trends.RBC.ru/trends/industry/67a0b8549a794784a5330840)

**Основные функции операционных систем**:

* **Управление памятью**. Операционная система распределяет ресурсы оперативной и постоянной памяти ПК и высвобождает её для различных процессов.
* **Управление процессором**. ОС распределяет вычислительные задачи между ядрами процессора, гарантируя, что каждое из них загружается равномерно.
* **Установка и обновление приложений и программ**. ОС предоставляет инструменты для безопасной и удобной установки программного обеспечения, а также контролирует процесс обновления.
* **Распределение потоков процессов и поддержка многозадачности**. Система эффективно управляет потоками выполнения задач, что позволяет одновременно запускать несколько приложений или процессов.

6.

**Компьютерная сеть** — это комплекс аппаратных и программных средств, предназначенных для обмена информацией и доступа пользователей к единым ресурсам сети. Основное назначение — обеспечить совместный доступ к информации (базам данных, документам и т. д.) и ресурсам (жёсткие диски, принтеры и т. д.). [slutskmedkol.by](https://slutskmedkol.by/Informac_tehnologii/LD/ТЗ%206.pdf)

По территориальному признаку сети делят на **локальные (LAN, Local Area Network)** и **глобальные (WAN, Wide Area Network)**.

### Локальная сеть (LAN)

**Объединяет устройства внутри ограниченного пространства** — дома, офиса или учебного заведения. Цель — обеспечить обмен данными и ресурсами между устройствами внутри сети. [digitalocean.ru](https://digitalocean.ru/n/chto-takoe-lokalnaya-set)[Skyeng.ru](https://Skyeng.ru/magazine/wiki/it-industriya/chto-takoe-lan/)

**Характеристики**:

* Устройства соединены между собой с помощью сетевых кабелей или беспроводных технологий (Wi-Fi).
* Топология может быть различной: звезда, кольцо, шина или смешанная.
* Для обмена данными используются различные сетевые протоколы, например TCP/IP.

[digitalocean.ru](https://digitalocean.ru/n/chto-takoe-lokalnaya-set)

**Сферы применения**:

* **Бизнес и предприятия** — локальные сети позволяют сотрудникам обмениваться информацией, работать в коллективе над проектами, использовать общие ресурсы.
* **Образование** — школы, колледжи и университеты используют локальные сети для связи между компьютерами в классах, дос**Функция в программировании** — это именованный блок кода, который выполняет одну конкретную задачу. Обычно функции выглядят как маленькие программы внутри основного кода, и их можно запускать в разные моменты времени. [studwork.ru](https://studwork.ru/spravochnik/programmirovanie/funkcii-v-programmirovanii-chto-eto-zachem-nujny-i-kak-rabotayut)[blog.skillfactory.ru](https://blog.skillfactory.ru/glossary/funktsiya/)

**Цель использования функций при написании программ** — **упорядочить и упростить процесс разработки**. Они позволяют разделить большой код на маленькие фрагменты — каждый отвечает за конкретное действие. [studwork.ru](https://studwork.ru/spravochnik/programmirovanie/funkcii-v-programmirovanii-chto-eto-zachem-nujny-i-kak-rabotayut)

**Некоторые преимущества использования функций**:

* **Повторное использование кода**. Вместо того чтобы копировать одни и те же строки много раз, можно написать функцию и вызывать её когда нужно.
* **Упрощение программы**. Код разбивается на логические части, его легче читать и поддерживать.
* **Изоляция задач**. Если в функции ошибка, её проще найти и исправить, не затрагивая весь код.
* тупа к онлайн-ресурсам и обмена информацией между преподавателями и студентами.

### Глобальная сеть (WAN)

**Объединяет устройства на больших географических расстояниях** — например, между городами или странами. Основная задача WAN — обеспечить стабильную и защищённую передачу данных между удалёнными узлами. [ru.vstack.com](https://ru.vstack.com/glossary/wide-area-network-wan/)[abimedia.ru](https://abimedia.ru/wan/)

**Характеристики**:

* Использует каналы связи различного типа: оптоволоконные линии, спутниковые каналы, магистральные сети провайдеров, мобильные каналы.
* В структуру WAN входят маршрутизаторы, коммутаторы, шлюзы и межсетевые экраны.

[ru.vstack.com](https://ru.vstack.com/glossary/wide-area-network-wan/)

**Сферы применения**:

* **Корпоративная сеть** — WAN связывает офисы в разных городах через VPN или MPLS.
* **Облачные сервисы** — обеспечивает доступ к удалённым серверам с минимальными задержками.
* **Интернет-провайдинг** — использует магистральные каналы для передачи данных между регионами.

[abimedia.ru](https://abimedia.ru/wan/)

**Отличия от LAN**:

* LAN работает в ограниченной зоне с высокой скоростью (до 10 Гбит/с), а WAN охватывает большие расстояния, но с меньшей скоростью (от 1 Мбит/с до 100 Гбит/с).
* В LAN управление обычно централизованное, а в WAN — распределённое между провайдерами и маршрутизаторами.

7. **Функция в программировании** — это именованный блок кода, который выполняет одну конкретную задачу. Обычно функции выглядят как маленькие программы внутри основного кода, и их можно запускать в разные моменты времени. [studwork.ru](https://studwork.ru/spravochnik/programmirovanie/funkcii-v-programmirovanii-chto-eto-zachem-nujny-i-kak-rabotayut)[blog.skillfactory.ru](https://blog.skillfactory.ru/glossary/funktsiya/)

**Цель использования функций при написании программ** — **упорядочить и упростить процесс разработки**. Они позволяют разделить большой код на маленькие фрагменты — каждый отвечает за конкретное действие. [studwork.ru](https://studwork.ru/spravochnik/programmirovanie/funkcii-v-programmirovanii-chto-eto-zachem-nujny-i-kak-rabotayut)

**Некоторые преимущества использования функций**:

* **Повторное использование кода**. Вместо того чтобы копировать одни и те же строки много раз, можно написать функцию и вызывать её когда нужно.
* **Упрощение программы**. Код разбивается на логические части, его легче читать и поддерживать.
* **Изоляция задач**. Если в функции ошибка, её проще найти и исправить, не затрагивая весь код.

**Git** — это система контроля версий, которая помогает отслеживать историю изменений в файлах. Её используют программисты для совместной работы над проектами. [blog.skillfactory.ru](https://blog.skillfactory.ru/glossary/git/)

**Некоторые задачи, для решения которых применяется Git:**

* **Возврат к предыдущей версии кода**. Это нужно, если решаемая задача больше не актуальна или требуется внести исправления в более раннюю версию программы. [blog.skillfactory.ru](https://blog.skillfactory.ru/glossary/git/)
* **Параллельная работа над проектом**. Несколько программистов одновременно вносят изменения в одно приложение или сайт, не мешая друг другу. [blog.skillfactory.ru](https://blog.skillfactory.ru/glossary/git/)[kotazzz.github.io](https://kotazzz.github.io/p/git-01/)[timeweb.com](https://timeweb.com/ru/community/articles/chto-takoe-git)
* **Объединение разных версий кода в один**. Например, над проектом трудится несколько программистов, и каждый разрабатывает или изменяет код в собственных ветках. В конце работы ветки сливают вместе — и получается цельная программа. [practicum.yandex.ru](https://practicum.yandex.ru/blog/chto-takoe-git-i-dlya-chego-nuzhen/)

8.

**Объектно-ориентированное программирование (ООП)** — это парадигма разработки программного обеспечения, согласно которой приложения состоят из объектов. Основная идея заключается в объединении данных и методов, работающих с этими данными, в единые структуры — объекты. [skillbox.ru](https://skillbox.ru/media/code/oop_chast_1_chto_takoe_klassy_i_obekty/)[sky.pro](https://sky.pro/wiki/python/osnovnye-ponyatiya-i-principy-oop/)

**Класс в ООП** — это **шаблон или чертёж для создания объектов**. Он определяет набор свойств и методов, которые будут у объектов этого класса. Например, класс «Персонаж» может иметь свойства «здоровье» и «уровень», а также методы «атака» и «защита». [sky.pro](https://sky.pro/wiki/python/osnovnye-ponyatiya-i-principy-oop/)

**Объект в ООП** — **это экземпляр класса**, который содержит данные и методы для работы с этими данными. Объекты могут взаимодействовать друг с другом, вызывая методы и изменяя состояние друг друга. Например, в игре объектом может быть персонаж, который имеет свойства (здоровье, уровень) и методы (атака, защита)

9.

**Git** — это система контроля версий, которая помогает отслеживать историю изменений в файлах. Её используют программисты для совместной работы над проектами. [blog.skillfactory.ru](https://blog.skillfactory.ru/glossary/git/)

**Некоторые задачи, для решения которых применяется Git:**

* **Возврат к предыдущей версии кода**. Это нужно, если решаемая задача больше не актуальна или требуется внести исправления в более раннюю версию программы. [blog.skillfactory.ru](https://blog.skillfactory.ru/glossary/git/)
* **Параллельная работа над проектом**. Несколько программистов одновременно вносят изменения в одно приложение или сайт, не мешая друг другу. [blog.skillfactory.ru](https://blog.skillfactory.ru/glossary/git/)[kotazzz.github.io](https://kotazzz.github.io/p/git-01/)[timeweb.com](https://timeweb.com/ru/community/articles/chto-takoe-git)
* **Объединение разных версий кода в один**. Например, над проектом трудится несколько программистов, и каждый разрабатывает или изменяет код в собственных ветках. В конце работы ветки сливают вместе — и получается цельная программа. [practicum.yandex.ru](https://practicum.yandex.ru/blog/chto-takoe-git-i-dlya-chego-nuzhen/)

10.

**Искусственный интеллект (ИИ)** в простом понимании — это набор технологий, которые позволяют собирать и анализировать данные, автоматизировать процессы и принимать обоснованные решения. [geeksforgeeks.org](https://tr-page.yandex.ru/translate?lang=en-ru&url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fartificial-intelligence%2Fai-in-agriculture-future-of-farming%2F)

**Некоторые примеры использования ИИ в сельском хозяйстве**:

* **Мониторинг сельскохозяйственных культур и управление ими**. Беспилотные летательные аппараты и спутники с ИИ предоставляют данные о состоянии здоровья культур, росте и состоянии почвы в режиме реального времени.
* **Прогнозная аналитика**. Алгоритмы ИИ обрабатывают исторические данные о погоде, состоянии почвы и урожайности для прогнозирования будущих тенденций. Например, модели могут предсказать оптимальные сроки посадки и сбора урожая, вероятные вспышки вредителей и ожидаемый урожай.
* **Автоматизированная техника**. Автономные тракторы и комбайны, управляемые ИИ, выполняют такие задачи, как посадка, прополка и сбор урожая с высокой точностью, снижая трудозатраты и сводя к минимуму человеческие ошибки.
* **Мониторинг состояния почвы**. Системы ИИ анализируют образцы почвы для определения уровней питательных веществ и других важных параметров.
* **Борьба с вредителями и болезнями**. Инструменты распознавания изображений на базе ИИ обнаруживают ранние признаки нашествия вредителей и болезней растений.
* **Мониторинг здоровья животных**. На тело животного устанавливаются специальные датчики, которые фиксируют температуру тела, сердечный ритм и активность. ИИ обрабатывает эти данные, что позволяет своевременно выявлять ранние признаки болезней и предотвращать их распространение.

11.